

Doelstellingen

Deze sessie gaat over hoe je Javascript kunt gebruiken om een pagina te manipuleren. In meer detail zijn de topics:

- Scripts kunnen koppelen aan een webpagina
- Document Object Model: kunnen vinden en manipuleren van elementen
- Kunnen werken met event listeners

Werkwijze, timing en upload

Dit labo loopt over één sessie en moet op het einde van de labosessie geupload worden via Toledo. Upload één enkele zip met daarin je oplossingen. De docent kan vragen de opdrachten thuis verder af te werken – je kunt dan wat extra punten verdienen. Neem die dan gewoon het volgende labo mee; je hoeft niet opnieuw te uploaden.

Het huishoudelijk reglement is van toepassing.

Opgaves

1. DOM oefenblad

Open de opgave voor het DOM oefenblad; de opgaves staan op het blad zelf. Probeer hier niet veel meer dan een uurtje aan te spenderen.

2. Formchecking

Open de opgave voor formchecking, en bestudeer de meegegeven HTML, CSS en Javascript code tot je het grondig begrijpt. Opgaves, in die volgorde uit te voeren:

1. Als bij de vraag over interesses ‘andere, namelijk...’ aangeklikt wordt, moet een tekstveld verschijnen (deze is reeds voorzien in de HTML code, maar verborgen met CSS). Als het weer aangeklikt wordt, moet het weer verdwijnen.
2. Voeg formchecking toe voor de naam.
3. Voeg formchecking toe voor het land.
4. Indien het formulier niet correct is ingevuld, moet de pagina naar boven scrollen (zoek op het Internet op hoe), en moet een melding bovenaan verschijnen (deze is reeds voorzien in de HTML code, maar verborgen met CSS).

5. Indien het formulier wel correct is ingevuld, moet het formulier verborgen worden, en de bedanktekst getoond worden (is reeds voorzien in de HTML code)
6. Gebruik de in CSS voorziene classes `invalid` en `valid` om bij formchecking te correct en incorrect ingevulde form controls te kleuren.
7. Voeg formchecking toe voor de drie slotvragen. Tip: je kunt met `querySelectorAll` en de juiste CSS selector alle aangevinkte checkboxes opvragen.
8. Voeg formchecking toe voor het e-mail adres; maak gebruik van een reguliere expressie om het e-mail adres te controleren (zoek op het Internet op hoe)
9. Voorzie de melding bovenaan van een kleine animatie als het getoond wordt. Maak gebruik van CSS transities en een CSS class om de melding te tonen (class toegewezen) of te verbergen (class niet toegewezen). Tip: gebruik de CSS properties `height` en `opacity`.
10. Om het helemaal af te maken: nadat een gebruiker de eerste keer het formulier heeft proberen in te dienen, moet live validatie ingeschakeld worden, d.w.z. dat de validatie bij elke verandering direct moet uitgevoerd worden. Baseer je op volgend schema om je code te herwerken:

```
;(function() {
  'use strict';

  // check while you type
  var liveCheck = false;

  // handles form checking
  var validateForm = function() {
    // formchecking here
    // ...
    return isValid;
  }

  // wait till DOM is loaded
  window.addEventListener('load', function() {
    // ...

    // live check while you fill the form
    document.getElementById('...').addEventListener('...', function() {
      if (!liveCheck) return;
      validateForm();
    });

    // intercept document submit
    document.getElementById('form1').addEventListener('submit', function(e)
  {
    // ...

    // check form
    var isValid = validateForm();

    // enable live check
    liveCheck = true;

    // draw conclusion
    // ...
  });
})();
```